Noname manuscript No.
(will be inserted by the editor)

Visual Simulation of Crack and Bend Generation in Deteriorated Films Coated on Metal Objects

—Combination of Static Fracture and Position-Based Deformation

Akinori Ishitobi · Masanori Nakayama · Issei Fujishiro

Received: date / Accepted: date

Abstract Weathering, an expression of degradation caused by rain and wind, is essential for photorealistic computer graphics. One of the commonest targets of weathering is metal, which is omnipresent in reality. However, for the realistic reproduction of scenes, many of which display degradation, the application of rust-proof paint to metal surfaces cannot be ignored. In our study, we propose a weathering method for coated films on metal objects, which are modeled using a three-dimensional (3D) triangular polygon mesh and deformed by combining two kinds of simulations: static simulation, for determining fractures based on the balance of the internal forces, and position-based bend simulation for moving vertices according to geometric constraints. Our method can digitally reproduce the deterioration of coated films using complex 3D deformation, which is difficult to achieve by material manipulation only.

Keywords Weathering · Coating · Deformation

1 Introduction

Age-related deterioration inevitably affects everything in reality. Changes to the appearance of deteriorated objects signify to our brain the passage of time, and the degree of deterioration is a meaningful factor suggesting the “age” of an object at a glance. Conversely, virtual worlds rendered by computer graphics (CG) differ from reality in that they are inherently unaffected by deterioration. Therefore, bridging this gap between the two worlds by assigning an “age” to CG objects necessitates the application of weathering, which expresses degradation caused by rain and wind.

Metallic corrosion is one of the commonest examples of real-life deterioration, and many weathering methods have been proposed. However, in reality, vulnerable metal is usually coated with rust-proof paint, so scenes in which metal alone is deteriorated are highly limited. In addition, most existing weathering methods focus on changing the material, while few deal with deformation. Some methods express three-dimensional (3D) fissures in rocks or the peeling of coated films based on the result of a crack simulation in a two-dimensional (2D) texture space, but the deformation remains superficial.

In response, Ishitobi et al. [10] proposed a deformation method of modeling coated films with a polygon mesh, but it focuses only on generating patterns of peeling; thus, it cannot generate long, branching cracks that split the coated film, nor can it simulate the process whereby the pieces split by cracks begin warping. As such, the aim of this study is to extend the method proposed by Ishitobi et al. [10] to simulate cracks and bends in coated films caused by deterioration. As shown in Fig. 1, this method enables the simulation of the 3D deformation of coated films and the generation of models difficult to design manually.

The remainder of this paper is structured as follows: the next section summarizes existing studies related to weathering and deformation simulation. Section 3 presents the procedure of modeling coated film deterioration and an overview of the entire simulation. Section 4 explains our method for simulating bent coated films, and Section 5 describes the criteria for generating fractures on coated films. Section 6 explains how to smudge the deformed model, while Section 7 shows the results of our simulation and Section 8 concludes this paper with brief remarks on future issues.

A. Ishitobi* · M. Nakayama · I. Fujishiro, Keio University,
*Corresponding author,
E-mail: akinori.ishitobi@fj.ics.keio.ac.jp.

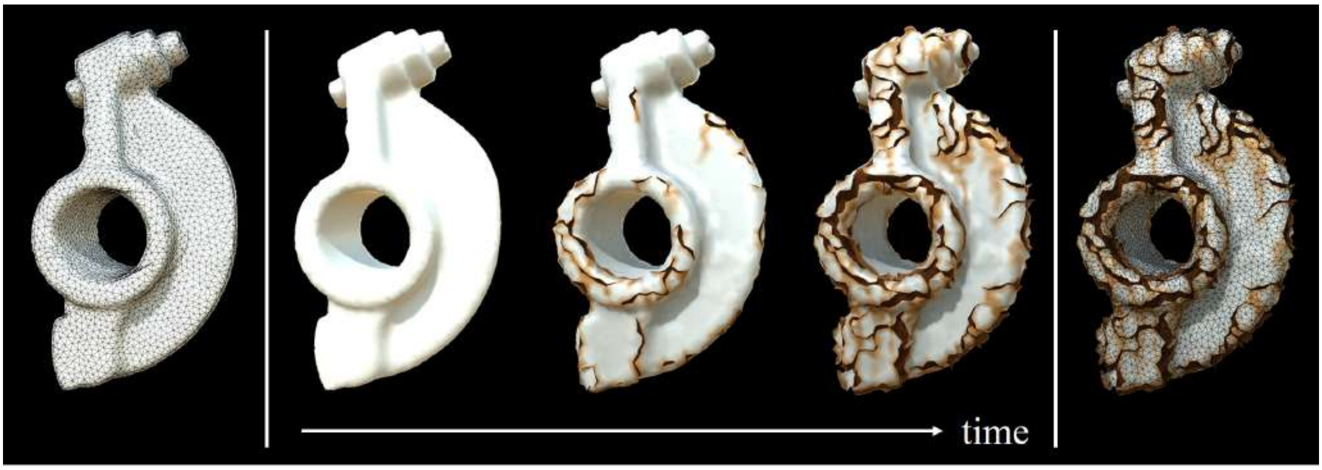


Fig. 1: Deterioration process of a mechanical part, simulated using the proposed method. The left- and right-most show the initial and final states of the mesh, respectively. The method does not deal with remeshing during the simulation, but it can generate cracks naturally by targeting a model with a uniformly random triangle mesh

2 Related work

In this section, we introduce weathering methods for expressing aged deterioration using CG. Then, we refer to simulation methods for solid deformation.

2.1 Weathering

Everything in reality is affected by age-related deterioration. The results of a survey by Merillou and Ghazanfarpour [18] demonstrate many known attempts to express changes in appearance due to aged deterioration. Specifically, weathering has been applied to various objects, including paper, corks [15], leaves [13], wood [16], stones [6] [30], and cities [21].

Metal has also been a primary target of CG weathering. The method for generating metallic patinas proposed by Dorsey and Hanrahany [7] is one of the most recognized weathering techniques, while metal corrosion [17] [11], the synthesis of rust in seawater [3], and the generation of rust texture [27] are other examples of metal weathering.

Conversely, few weathering methods have targeted coated objects. Paquette et al. [22] and Gobron et al. [8] demonstrate the peeling of coated films by generating cracks and bending in the surrounding films. However, cracks are simulated in the texture space, while bending is simulated using a 3D polygon mesh, so combining both processes is not straightforward. As a result, only coated films near cracks bend, and large warps cannot be expressed. Ishitobi et al. [10] simulated both cracks and peeling in a 3D polygon mesh, though their method focuses only on generating patterns of peeled areas, and cracks and out-of-plane-direction deformations appear only in nearby pealed areas. Therefore, the method cannot express such peeling phenomena, often observed in reality as branched, complex cracks with large warping.

As examples of versatile weathering, estimation of the deterioration process with an appearance manifold [29] [31], the generation of a time-variant texture [1], and the weathering of a single image [9] have been proposed. However, these methods manipulate the material of object surfaces or objects in images, but they do not enable 3D deformation. In response, γ-ton tracing [5] is a generalized method introduced to simulate the distribution of the degree of aged deterioration throughout the 3D space, but it must be combined with other techniques to express spatial deterioration.

2.2 Deformation simulation

The finite element method (FEM) [28] is a representative simulation method applicable to solid deformation. When it is applied to structural analysis in the field of physical simulation, an element equation can be formulated for each element constituting an object. The solution to each equation is the displacement at the nodes located on the surface of or inside the element. In the field of CG, many deformation methods use FEM. Among them, the simulation of brittle fractures by O'Brien [20] is one of the commonest attempts. Alternatively, peridynamics [24] and the material point method [25] are known as mesh-free, particle-based approaches to producing fractures.

Deformation methods based on continuum mechanics, including FEM and peridynamics, lead to exact results according to the governing equations, but applying them to complex phenomena is difficult, because they require the formulation of governing equations. Conversely, position-based dynamics (PBD) [19], a well-known CG simulation method, replicates deformation and motion based on geometric constraints instead of natural laws. PBD enables the easy simulation of complex mechanical phenomena, because the values of the geometric parameters produced as a result of a phenomenon can be given as constraints. However, PBD may lead to physically incorrect behavior, but Bender et al. [2] realized a physically exact PBD by setting constraints based on mechanical energy.

Aged deterioration progresses under the influence of various complexly intertwined factors, so it is difficult to formulate governing equations, even if considering only the factors necessary to reproduce the appearance plausibly. In addition, when simulating the deformation of a thin board, such as coated films, by FEM, the computation is more complicated, because we must derive not only the displacement but also the angles of deflection as the solution. Particle methods must position multiple particles in the thickness direction and require much computational complexity. In both FEM and particle methods, changes to the values of some mechanical parameters due to aged deterioration cannot be theoretically derived or actually measured. Thus, the proposed method expresses bends in coated films using PBD-based deformation and realizes natural deformation by setting geometric constraints based on a static simulation.

3 Method outline

In this section, we outline our method, which mimics the aged deterioration of coated films in reality. We also introduce a basic model to reproduce cracks and bends, which are difficult to depict by physical simulation, as explained in Section 2.2. Note that the simplification is not intended to neglect complex factors but, conversely, to establish a policy within which they can be incorporated.

3.1 Underlying mechanics

The main factor in the aged degradation of real coated films is the change to the internal stress and physical properties due to drying. In reality, because it is difficult to measure stress in coated films, we compute it from measurable displacement by coating a small metal plate and letting the base deform together with the coated film. According to Stoney's law [26], when a coated, thin metal disk of radius r, thickness d_S, Poisson's ratio ν_S, and Young's modulus E_S bends, the relation between the stress in the coated film σ_F and the radius of the curvature R can be expressed by:

$$\sigma_\mathrm{F} = \frac{E_\mathrm{S} {d_\mathrm{S}}^2}{6(1-\nu_\mathrm{S})d_\mathrm{F}} R^{-1}, \tag{1}$$

where d_F denotes the thickness of the coated film. If the center of the disk is fixed, R is approximated as:

$$R = \frac{r^2}{2\delta},$$

where δ denotes the displacement of the circumference. Substituting it into Eq. 1, we obtain:

$$\delta = \frac{3(1-\nu_\mathrm{S})d_\mathrm{F}}{E_\mathrm{S} {d_\mathrm{S}}^2} \sigma_\mathrm{F} r^2.$$

Then, multiplying σ_F by the volume of the coated film $\pi r^2 d_\mathrm{F}$ yields the contraction force of the coated film $T = \pi r^2 d_\mathrm{F} \sigma_\mathrm{F}$, and we obtain the relation between T and δ as follows:

$$\delta = \frac{3(1-\nu_\mathrm{S})}{\pi E_\mathrm{S} {d_\mathrm{S}}^2} T. \tag{2}$$

If the contraction force becomes strong enough, the coated film tears and a crack appears. In addition, the bending force peels the film from the base.

In general, there intrinsically exist tiny cracks in the material as defects, and they grow into visible cracks. At an end of a crack, internal stress concentrates, causing a material fracture and extending the crack. Cracks tend to grow straight because the stress is especially strong in the direction of crack propagation.

Aged deterioration, including the peeling of coated films, progresses gradually for a long period, so it usually appears stationary during short observations. In other words, it is regarded as a quasi-static process, where the state changes slowly while maintaining equilibrium. While it usually takes a decade for coated films to begin peeling, it only takes a short period for a piece of peeled film to fall off an object, so its process should not be described in the simulation of aged deterioration.

3.2 Simulation flow

In the proposed method, a coated film is modeled with a triangle polygon mesh. As shown in Fig. 2, peeling due to aged deterioration progresses in three stages: separation, tearing, and stripping. In this paper, the transition to the next stage is referred to as *fracturing.* Note that these stages can coexist in the same space, because a fracture is judged for each local region and is based on a static equilibrium. The parameter values

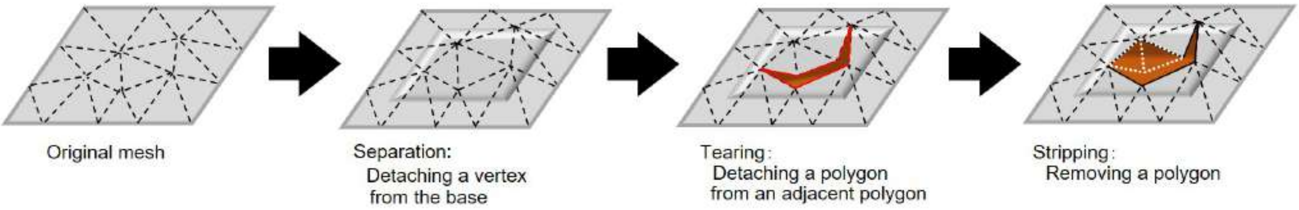

Original mesh
Separation:
Detaching a vertex from the base
Tearing:
Detaching a polygon from an adjacent polygon
Stripping:
Removing a polygon

Fig. 2: Peeling process in the proposed method. Whether to progress to the next stage is determined by the static fracture criteria

and phase information of the polygon mesh are updated when a part of the model fractures, the specific computation method and process of which will be detailed in Section 5.

Moreover, the position-based bend simulation described in Section 4 parallels static simulation, where the bend simulation proceeds following a coherent method independent of the fracture stage and its target is updated when moving on to the next stage.

Further, we regard a smudge as non-deformation aged deterioration because it only affects the color of each vertex. We explain the details of smudges in Section 6.

3.3 Basic model

To reproduce the peeling of coated films using a polygon mesh, we introduce a basic model, as shown in Fig. 3. In this model, each polygon applies bending and contraction forces to its adjacent polygons, while each node attaches to the base in the initial state, but its adhesion force may become weak and may separate from the base over time. The contraction force becomes strong during the simulation, and the connection of two adjacent polygons is torn when the contraction force becomes stronger than the binding force of the polygons.

The basic model is implemented with a half-edge structure (Fig. 4), which consists of three kinds of geometric elements: vertices, faces, and half-edges. Mechanical parameters are set to each geometric element, as follows. Adhesion force to the base is set to the vertices, contraction force is set at the faces, and binding force and bending force are set to the half-edges. All four parameters take scalar values. For simplicity, we refer to a pair of adjacent half-edges simply as an edge.

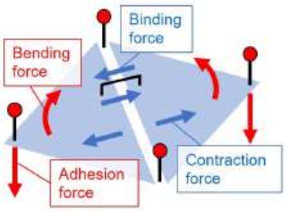


Fig. 3: Basic model. Red and blue arrows indicate forces compared for judging separation and tearing, respectively

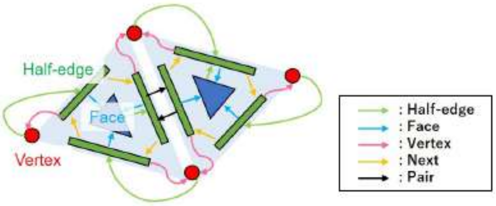


Fig. 4: Half-edge structure. Each arrow means the elements can be tracked in its specified direction

4 Bend simulation

This section introduces a simulation method for bent films. By applying it to a coated film with cracks, the film on each side of the crack will bend and the cracks will widen.

4.1 Position-based deformation

We begin with the outline of PBD [19], which is the basis of our bend simulation. In PBD, geometric constraints are set on the vertices in a scene, and the positions of the vertices are repeatedly updated to satisfy the constraints. A constraint on N vertices is described as a function of their position vectors $C(\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_N)$, and the function is defined such that its value is 0 when the constraint is satisfied. Then, for the i-th $(i = 1, 2, ..., N)$ vertex, if its inverse mass is w_i, its correction for the position $\boldsymbol{p}_i$ is computed by:

$$\begin{aligned} \varDelta\boldsymbol{p}_i &= -sw_i \boldsymbol{\nabla}_{\boldsymbol{p}_i} C(\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_N), \\ s &= \frac{C(\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_N)}{\sum_j w_j \|\boldsymbol{\nabla}_{\boldsymbol{p}_j} C(\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_N)\|^2}. \end{aligned} \tag{3}$$

As can be seen from Eq. 3, the corrections are computed from the gradient of C with respect to the vertex positions. Therefore, C must be differentiable and have no extrema within the domain of the definition.

PBD predicts rough positions in the next step by considering inertia, and it then modifies them using the above equation. However, because aged degradation is regarded as a quasi-static process, we directly modify the positions without considering inertia. In this paper, such a version of PBD will be referred to as PBD-2. Note that the constraints explained in Sections 4.2 and 4.3 are applicable to PBD-2, as well as to the original PBD method.

4.2 Length constraint

For an edge whose ends are $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$, a constraint $_{\text{length}}(\boldsymbol{p}_1, \boldsymbol{p}_2)$ to maintain its length at a constant value d_0 is defined as:

$$C_{\text{length}}(\boldsymbol{p}_1, \boldsymbol{p}_2) = \|\boldsymbol{p}_1 - \boldsymbol{p}_2\| - d_0.$$

Substituting it into Eq. 3 yields:

$$\begin{aligned} \varDelta\boldsymbol{p}_1 &= -\frac{w_1}{w_1 + w_2}(\|\boldsymbol{p}_1 - \boldsymbol{p}_2\| - d_0)\frac{\boldsymbol{p}_1 - \boldsymbol{p}_2}{\|\boldsymbol{p}_1 - \boldsymbol{p}_2\|}, \\ \varDelta\boldsymbol{p}_2 &= +\frac{w_2}{w_1 + w_2}(\|\boldsymbol{p}_1 - \boldsymbol{p}_2\| - d_0)\frac{\boldsymbol{p}_1 - \boldsymbol{p}_2}{\|\boldsymbol{p}_1 - \boldsymbol{p}_2\|}. \end{aligned}$$

4.3 Bend constraint

Constraints for angle control are also proposed by Müller et al. [19], but they cannot be applied to our method, which needs to bend the mesh model. To explain why, we first explain the angle constraints proposed by Müller et al. As shown in Fig. 5(a), we consider a constraint to maintain the angle ϕ formed by adjacent faces at a constant value ϕ_0. Let $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ be the end positions of the edge shared by the faces and $\boldsymbol{p}_3$ and $\boldsymbol{p}_4$ the positions of the other vertices. Let us assume that $\boldsymbol{N}_1 = (\boldsymbol{p}_1 - \boldsymbol{p}_2) \times (\boldsymbol{p}_2 - \boldsymbol{p}_3)$ and $\boldsymbol{N}_2 = -(\boldsymbol{p}_1 - \boldsymbol{p}_2) \times (\boldsymbol{p}_2 - \boldsymbol{p}_4)$ are directed to the front side of each face. By using the unit normal vectors of each face $\boldsymbol{n}_1 = \boldsymbol{N}_1/\|\boldsymbol{N}_1\|, \boldsymbol{n}_2 = \boldsymbol{N}_2/\|\boldsymbol{N}_2\|$, ϕ is given by $\phi = \text{acos}(-\boldsymbol{n}_1 \cdot \boldsymbol{n}_2)$. Further, the angle constraint is defined as:

$$C_{\text{angle}}(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \boldsymbol{p}_4) = \phi - \phi_0.$$

Substituting this equation into Eq. 3 yields corrections for the positions, as detailed in Müller et al. [19], but for reasons discussed below, we do not use the C_{angle}.

The domain of ϕ is $[0, \pi]$, and it cannot distinguish between mountain-fold and valley-fold. As such, if we set the value of ϕ_0 to less than π, the faces may bend in the opposite direction than intended. Moreover, the C_{angle} is not differentiable when $\phi = \pi$, and the corrections are computed considering a limit value of $\mathbf{0}$. Therefore, when two polygons are flat, C_{angle} cannot deform them. The atan2 function can be used instead of acos to distinguish between valley and mountain folds, but it is not differentiable when $\phi = \pi/2$ and $\phi = 3\pi/2$, and it cannot compute corrections.

Physically based methods considering bending energy, such as in [4], realize a qualitative thin-shell simulation, where the bend direction can be determined, but it requires more computational time than position-based methods. Jeong et al. [12] proposed a simulation method for sheet bending in a certain direction by structuring two-layer meshes, but it cannot be applied to polygon mesh models with irregular grids.

In our method, as shown in Fig. 5, let $\boldsymbol{n}_{\text{neutral}} = (\boldsymbol{n}_1 + \boldsymbol{n}_2)/\|\boldsymbol{n}_1 + \boldsymbol{n}_2\|$ be a neutral direction vector and ϕ_1, ϕ_2 angles between $\boldsymbol{n}_{\text{neutral}}$ and the faces. Two constraints are set such that both ϕ_1 and ϕ_2 approach $\phi_0/2$. By holding the two angle constraints, we can designate the bend direction as $0 < \phi_0 < \pi$ for a valley fold and $\pi < \phi_0 < 2\pi$ for a mountain fold.

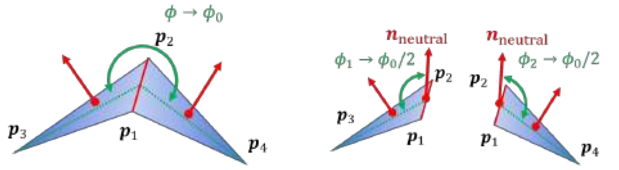
Hereinafter, we refer to these divided angle constraints as *bending constraints.* The C_{angle} is certainly applicable as an angle constraint, but we adopted a simple linear function based on a triangle bending constraint [14] to avoid the numerical inaccuracy of the acos function. The application of the triangle bending constraint is detailed in the Appendix.

By setting constraints to preserve the length of each edge and bending constraints to narrow the angle between every two adjacent faces, the polygon mesh bends in a certain direction, as shown in Fig. 6.

5 Link between static simulation and bend simulation

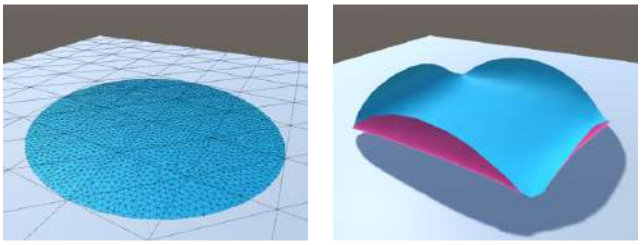
In the proposed method, a fracture is judged based on the mechanics theory described in Section 3.1. As shown in Fig. 7, by updating the constraints of PBD-2 when a fracture occurs, the bend simulation is applied to aged coated films.

If we update positions considering only geometric constraints, coated films might sometimes sink into the face. General PBD detects collisions and sets constraints for resolving the collisions to prevent objects



(a) Before division (b) After division

Fig. 5: Division of an angle constraint. The neutral direction vector $\boldsymbol{n}_{\text{neutral}}$ is decided based on the normal vectors of the faces and the angles ϕ_1, ϕ_2 between $\boldsymbol{n}_{\text{neutral}}$ and the faces, which are set as the target of the constraints



(a) Mesh at the initial state (b) Simulation result

Fig. 6: Bend simulation on a disk. The number of polygons is 4,047, and the target value of angles formed by every two adjacent polygons is set to 10°

from sinking, but in our method, we adopted simplified collision detection, where the motion space of a vertex is limited to the outer region from its initial position. The boundary of the motion space is the plane perpendicular to the initial normal, and it contains the initial position.

5.1 Separation

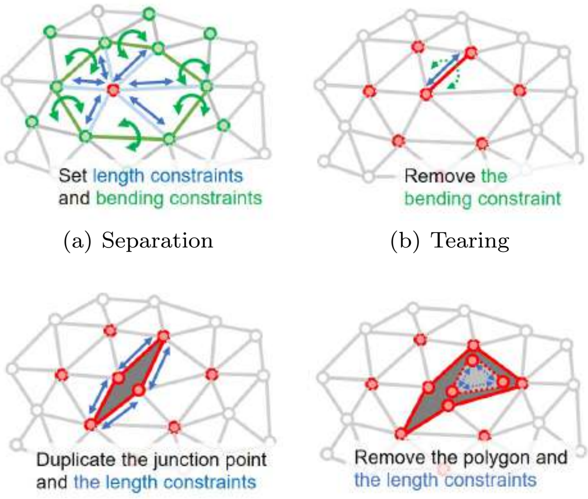
The distribution of adhesion force F_{A} in the initial state is generated by Perlin noise [23] to reproduce a realistic output that is uneven and continuous. F_{A} at each vertex linearly decreases according to:

$$F_{\mathrm{A}}^{n+1} = \begin{cases} F_{\mathrm{A}}^{n} - w\varDelta t & (F_{\mathrm{A}}^{n} > 0) \\ 0 & (F_{\mathrm{A}}^{n} \leq 0) \end{cases},$$

where w denotes the speed of aged deterioration and $\varDelta t$ the time step.

Conversely, F_{B} is a virtual force acting to peel the coated film from the base. Assuming F_{B} equals the force required to bend coated films with the base when the based is not fixed, F_{B} can be computed according to Eq. 2. If F_{B} is proportional to δ in accordance with Hooke's law and all parameters other than T are constants, F_{B} is simply given by:

$$F_{\mathrm{B}} = kT,$$



(a) Separation (b) Tearing

(c) When a vertex is duplicated as a result of tearing (d) Stripping

Fig. 7: Constraint processing at fracturing. Separated vertices and torn edges are shown in red. Blue and green arrows indicate length and bending constraints, respectively

where k denotes a constant coefficient that represents the ease of separation. The series of computations explained in Section 3.1 assumes the base is flat in the initial state. However, when observing actual coated films on curved objects, convex parts appear to peel easily. To reproduce this kind of curvature effect, we multiply the bending force by the value of the sigmoid function S, whose input is the angle formed by adjacent faces ϕ $(0 < \phi < 2\pi)$:

$$F_{\mathrm{B}} = kTS(\phi) = \frac{kT}{1 + e^{-\gamma(\phi - \pi)}}, \tag{4}$$

where γ is a positive constant. The larger its value, the greater the effect of the curvature on the ease of peeling. As shown in Fig. 8(a), if $\gamma = 1$, the effect of the curvature is so small that aged deterioration uniformly progresses even on the steep curve at the bottom of the model. Alternatively, as shown in Fig. 8(c), if $\gamma = 4$, the effect of the curvature is so great that the difference in the degree of aged deterioration appears even on the gentle curve at the top of the model. Thus, we adopted $\gamma = 2$, where the difference in the degree of aged deterioration appears only on somewhat steep curves, as in Fig. 8(b).

If a vertex separates, we set bending constraints on pairs of adjacent faces whose shared edges exert a bending force onto the vertex, and the target value of the angle is narrowed gradually. Simultaneously, constraints to preserve the initial length are set on each edge connected to the vertex.

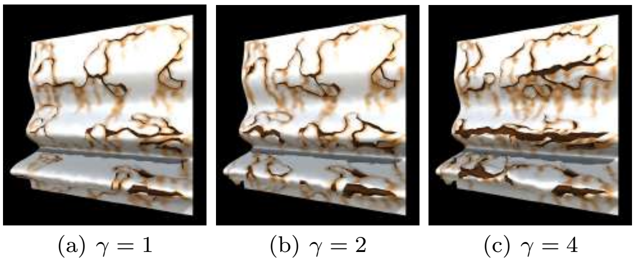

(a) $\gamma = 1$ (b) $\gamma = 2$ (c) $\gamma = 4$

Fig. 8: Sensitivity analysis of the geometric property γ in Eq. 4, which controls the effect of the curvature on the ease of peeling

5.2 Tearing

If both ends of an edge are separated and the binding force is less than the sum of the contraction force of two adjusted faces, the edge tears. To reproduce the direction dependency of cracks, we introduce a crack that extends easily along the direction vector of the crack, which is a unit vector that runs to the end of the crack from the torn edge. If either of an edge's ends is also an end of an existing crack, and if the inner product of the unit vector from the end of the crack to the other end of the edge, and if the direction vector of the crack is more than a constant α $(-1 < \alpha < 1)$, the edge will tear easily. In this case, the product of the binding force and its decreasing rate β $(0 < \beta < 1)$ are compared instead of the binding force to judge tearing. Thus, we empirically set $\alpha = 0.45$ and $\beta = 0.20$.

While the contraction force is initially set uniformly for each face, the binding force is set to be extremely small on certain edges to reproduce defects. We empirically set the existence probability of the weak edges to 1 % and the binding force of weak edges to 0.1 % of the other edges. When an edge is torn, the contraction forces of the two faces next to the torn edge are decreased. If a face has a torn edge, the binding force of the other edges decreases continuously.

The bending constraint set on the faces next to the torn edge is removed, and the following process is performed on each of its end vertices. If only one edge is torn among the edges connected to the vertex, that is, if the vertex is an endpoint of the crack, the direction vector of the crack is recorded. Conversely, if another torn edge exists among those connected to the vertex, that is, if the vertex is a junction of the cracks, it is duplicated at the same position, and phase information and constraints are updated to prevent referring vertices across cracks. In this case, the length constraint shared by the torn half-edges is also duplicated, such that each half-edge has its own length constraint.

5.3 Stripping

When all half-edges surrounding a face are torn, the face is stripped off, and it is removed along with its surrounding half-edges. Note that this is the only case in which geometric elements are removed. Then, length constraints on the removed half-edges are also removed. Because the bending constraints on the stripped face are removed at tearing, all unnecessary constraints related to the face are completely eliminated.

6 Stain expression

Aged deterioration of coated films causes not only deformation but also color and material changes. This section describes the method for depicting rust run-off and black stains, respecting the method of Ishitobi et al. [10].

6.1 Rust run-off stains

The amount of rust is recorded for each of the vertices and propagated downward from peeled areas along the object's surface. Thus, the color and reflectance at a vertex are changed depending on the amount of rust.

In this method, run-off must begin when coated films are torn, because cracks are wide enough for rust to flow out via the bend simulation in Section 4. To prevent rust from flowing from the entire crack outline, we classify vertices on the outlines into a source or non-source of rust and allow rust to flow only from the former. If an edge's ends are vertex A and vertex B, whether A is a source is determined according to the following criteria. Note that if A is duplicated, the duplicate A' is located above A, so A' is designated a non-source.

1. If A is the end of a crack and below B, A is designated a source (Fig. 9(a))
2. If A is the end of a crack and above B, A is designated a non-source (Fig. 9(b))

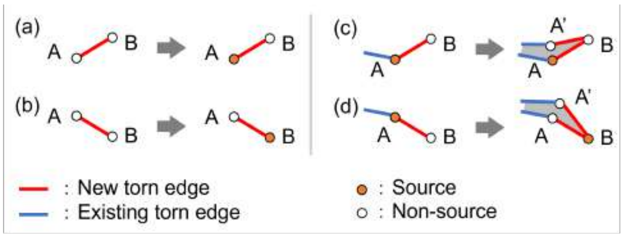


Fig. 9: Setting of rust sources at tearing

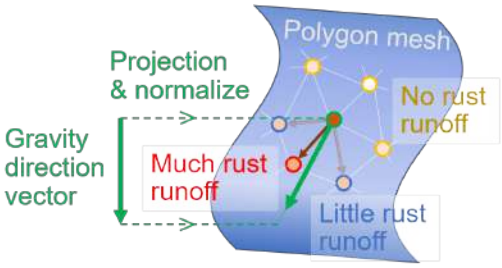


Fig. 10: Overview of rust propagation. Rust flows downward along the surface

3. If A is not the end of a crack and below B, A is not changed (Fig. 9(c))
4. If A is not the end of a crack and above B, A is set to a non-source (Fig. 9(d))

This process places the rust source lower than nearby vertices on the crack, even if the crack is extended and branched.

Because external factors, such as wind, often alter flow paths, over the long term, rust diffuses and propagates downward. The direction of rust propagation is determined by projecting the gravity direction onto the model surface, as illustrated in Fig. 10. If a vertex has rust, the amount of rust at a nearby vertex in the projected gravity direction is increased. Detailed algorithms with variable values follow Ishitobi et al. [10].

6.2 Black stains

The amount of dust is computed at each vertex, and the material of the entire model is changed by darkening the color and decreasing the reflectance at vertices, depending on the dust amounts. While black stains are influenced by physical and chemical effects in reality, the speed of dust accumulation in this simulation method is computed only from geometric quantities to generate a plausible dust distribution effectively.

The geometric quantities referenced to compute black stains include height, direction, and curvature. At a vertex, the higher its height, the closer its normal direction to that of gravity, and the more convex its curvature, the slower dust accumulates there. Figure 11 shows the results of considering each factor and the result of considering all three factors.

7 Results

We used a PC with an Intel Core i9-10980 CPU, Intel UHD Graphics, and 32.0 GB RAM, and we ran simulations using Unity 2019.4.14f1.

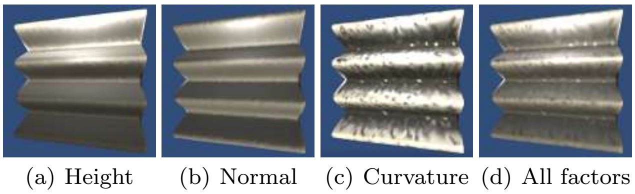

(a) Height (b) Normal (c) Curvature (d) All factors

Fig. 11: Geometric factors and appearance of black stains

Figure 12 compares the results of weathering using the method by Ishitobi et al. [10] and using our method. While cracks appear only near peeled areas when using the method of Ishitobi et al., our static simulation generates long, branching cracks. In addition, our bend simulation expresses the warping of coated films and produces harmony between widened cracks and peeled areas.

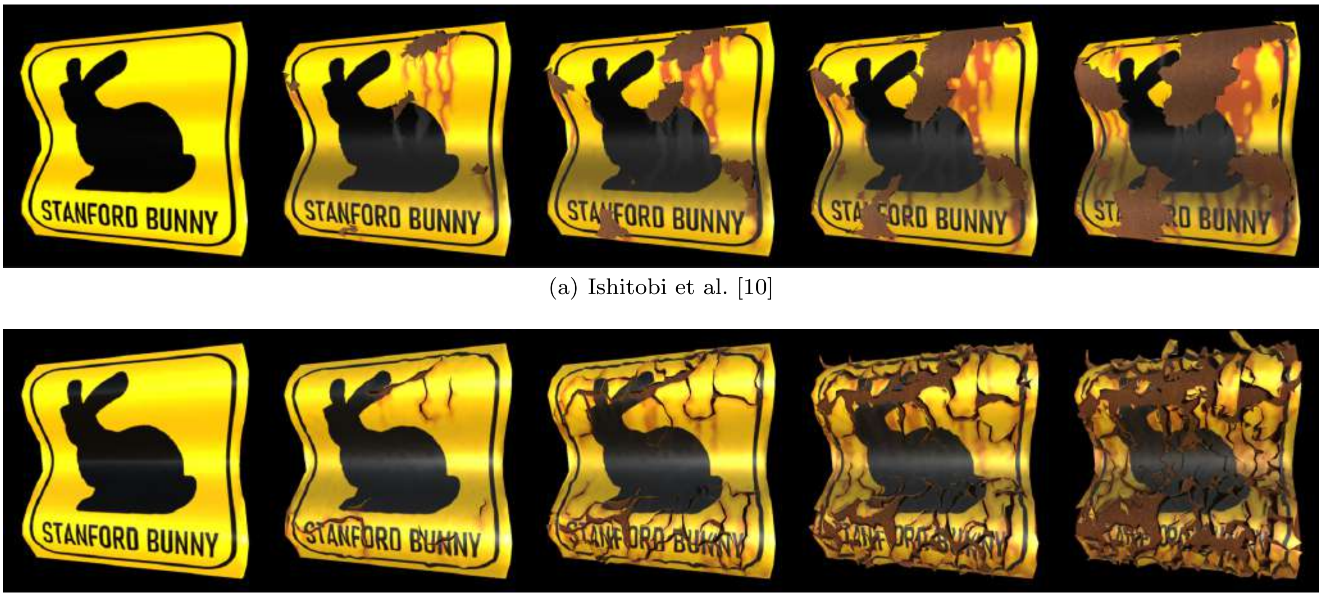
Figure 13 compares a snapshot of a weathered object with simulation results from using the method by Ishitobi et al. [10] and using our method with a bend simulation. This example displays cracks and warps generated by our method, reinforcing the reality of aged deterioration.

Another result of applying our method with a bend simulation to a machine part model is shown in Fig. 1. This example proves our method is applicable to a model of such a complex geometry.

Figure 14 shows a weathered object embedded in a specific scene. In many realistic scenes, in which most other objects are more or less aged, weathering harmonizes the object and the environment, and it reinforces the reality of the object.

We applied our method with a bend simulation to three models: a flat plate, a curved plate, and a Stanford bunny, the results of which are shown in Fig. 15, and the statistics of the simulation are listed in Table 1. Computation time increases with the number of polygons, but even the model with 10^4 polygons can be simulated at about 10 frames per second. The computation time ratios of each listed process are almost equal among the three cases, with the static simulation and bend simulation ratios constituting 35 % and 45 %, respectively.

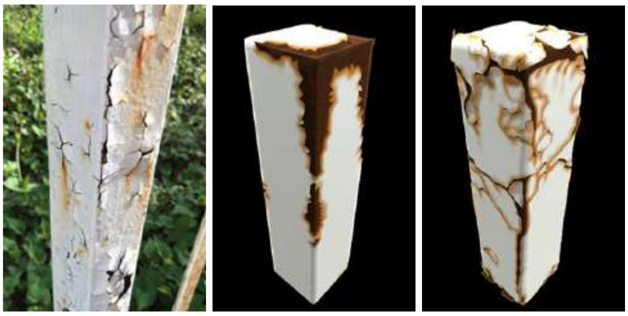
Figure 16 shows a weathering process with an external input that can visualize the areas of a simulated model likely to fracture by coloring them according to the difference between the adhesion force and lifting force. Partially varying the adhesion and binding force values allows us to control the weathering progress interactively. Refer to the supplemental video for more details on the directable control.



(a) Ishitobi et al. [10]

(b) Our method

Fig. 12: Comparison of the method of Ishitobi et al. [10] (a) and our method (b). Adjusting the control parameters could allow our method to visualize aged coated objects with few cracks, such as in (a), but we focus herein on the ability of the proposed method to generate striking cracks and bending, as shown in (b)



(a) (b) (c)

Fig. 13: Comparison of a snapshot of a deteriorated coated object (a), the result of a simulation with the method of Ishitobi et al. [10] (b), and the result of our simulation (c)

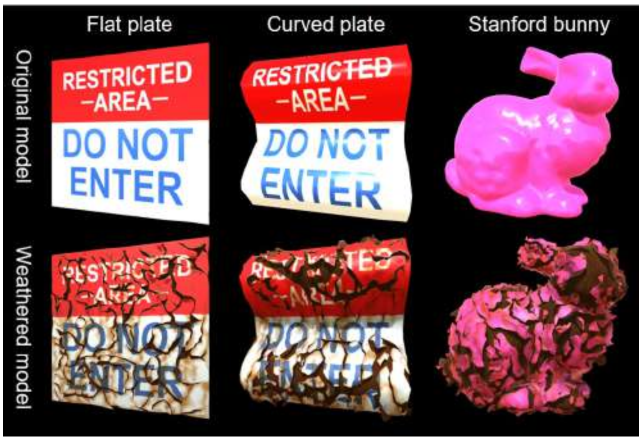


Fig. 15: Results of weathering three kinds of models. Associated statistics are available in Table 1

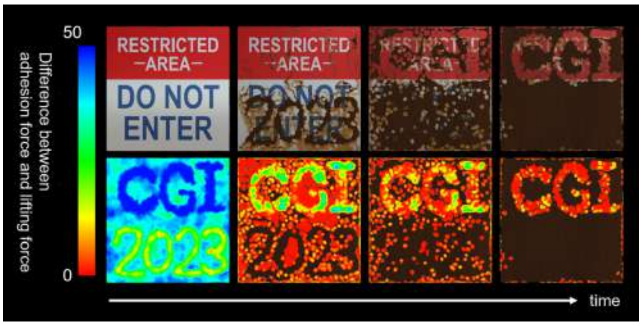
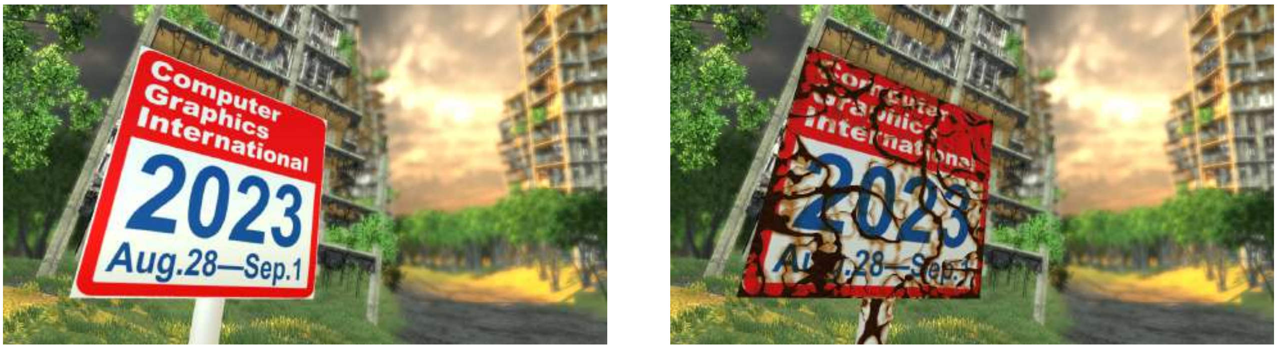


Fig. 16: Result of weathering simulation with intended control (upper row) and the visualization of areas likely to fracture (bottom row)

8 Conclusion

In this paper, we expressed the deformation of coated films on metal objects due to aged deterioration by combining a static simulation and a bend simulation. In the static simulation, fractures based on a simple computation of force balance and the phase manipulation of the polygon mesh were able to generate cracks naturally. In the bend simulation, we applied PBD-based deformation to coated films to reproduce realistic peeling. In addition, we depicted rust run-off stains and black stains considering geometric properties to render the appearance of the coated film models as more natural.



(a) Before (b) After

Fig. 14: Weathering of a sign model in a devastated city scene. Weathering integrates the object into the scene

Our method does not deal with remeshing, so the simulation results depend only on the fineness of the input mesh. The results of applying our method to models with different fineness values are shown in Fig. 17, where input model in (a) has 3,012 polygons, while both models in (b) and (c) have 11,488 polygons. The models in Fig. 17(a) and Fig. 17(b) are simulated with the same parameter settings, except for the fineness, but the model in Fig. 17(b) generates more complex cracks whose densities are uneven. Cracks of different fineness values, as shown in Fig. 17(a) and (b), coexist on a poorly tessellated mesh, so we set a limitation that the mesh must be uniform. Conversely, the model in Fig. 17(c) set parameter α, mentioned in Section 5.2, to a higher value to avoid a fine crack and to generate cracks with the same complexity as in Fig. 17(a). Thus, we should optimize parameters, such as α, by considering physical property values and mesh size.

In the results shown thus far, though the mesh used for the simulation is shown, it is possible to use another fine mesh for rendering. Figure 18 shows the result of a simulation displayed on a high-resolution model by mapping the plate model with 3,012 polygons, as shown in Fig. 15, to a fine mesh with 46,399 polygons. Mesh mapping was precomputed, and the mesh for rendering was generated from the simulation mesh and mesh mapping. Compared to the original method, which conducted both simulation and rendering using the coarse mesh, the precomputation time increased by a factor of 9, and the computation time per frame took 4 times longer. Note that when generating the mesh for rendering, the state of the previous step need not be maintained, so the fine mesh can be displayed only when needed; otherwise, the coarse mesh is normally shown.

Other future works should consider additional weathering factors. Particularly, moisture should be the first target, as it significantly influences the physical properties of coated films, stains, and base corrosion.

Table 1: Computation time and its breakdown when weathering the flat plate, curved plate, and Stanford bunny in Fig. 15. The items of the breakdown are, from top to bottom, static simulation and setting of constraints, bend simulation, generation of mesh for display, expression of stains, and others

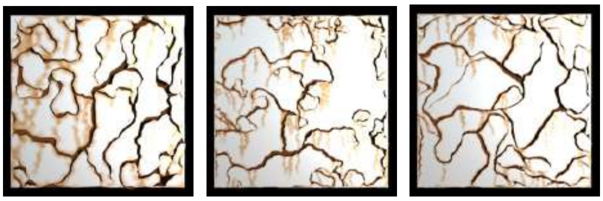
	Flat plate	Curved plate	Stanford Bunny
Steps	1,000	1,000	1,000
Polygons	3,012	3,005	10,000
Time (s)	15.4	16.7	65.8
Breakdown of computation time (%)			
Fracture	46.7	44.4	45.4
Deformation	33.7	36.8	36.7
Mesh generation	13.3	12.4	11.7
Stains	4.7	4.8	4.6
Others	1.6	1.6	1.6

Conflict of interest: The authors declare that they have no conflict of interest.

Data availability statement: The data related to this study is available from the corresponding author upon request.

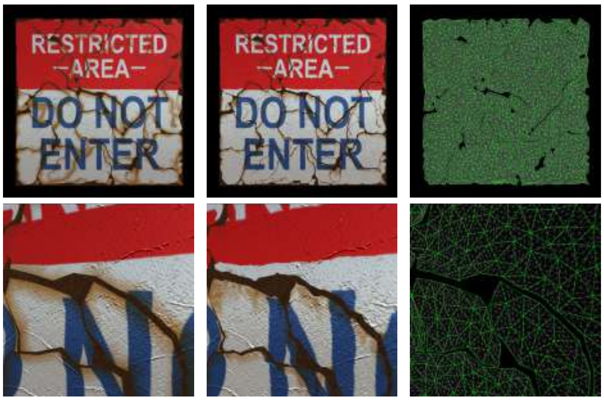
References

1. Bellini, R., Kleiman, Y., Cohen-Or, D.: Time-varying weathering in texture space. ACM Transactions on Graphics **35**(4), 141:1–141:11 (2016)
2. Bender, J., Koschier, D., Charrier, P., Weber, D.: Position-based simulation of continuous materials. Computers & Graphics **44**, 1–10 (2014)
3. Chang, Y.X., Shih, Z.C.: The synthesis of rust in seawater. The Visual Computer **19**(1), 50–66 (2003)
4. Chen, H.Y., Sastry, A., van Rees, W.M., Vouga, E.: Physical simulation of environmentally induced thin shell deformation. ACM Transactions of Graphics **37**(4), 146:1–146:13 (2018)
5. Chen, Y., Xia, L., Wong, T.T., Tong, X., Bao, H., Guo, B., Shum, H.Y.: Visual simulation of weathering by γ-ton tracing. In: Proceedings of ACM SIGGRAPH 2005, pp. 1127–1133 (2005)



(a) 3,012 polygons, $\alpha = 0.45$ (b) 11,488 polygons, $\alpha = 0.45$ (c) 11,488 polygons, $\alpha = 0.77$

Fig. 17: Results of weathering square polygon mesh models with different fineness values. Each square has the same side lengths. The finer mesh in (b) generates more complex cracks than that in (a). By increasing the parameter value to control the crack direction α, the finer model (c) can maintain the same complexity as that of the cracks in (a)



(a) Mesh for simulation (b) Mesh for rendering (c) Mesh in wireframe

Fig. 18: Expression of fine cracks using two different meshes: A coarse mesh for simulation with 3,012 polygons in (a) and a fine mesh for rendering with 46,399 polygons in (b). (c) shows the coarse and fine mesh wireframes in green and gray, respectively

6. Desbenoit, B., Galin, E., Akkouche, S.: Modeling cracks and fractures. The Visual Computer **21**(8), 717–726 (2005)
7. Dorsey, J., Hanrahan, P.: Modeling and rendering of metallic patinas. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, pp. 387–396 (1996)
8. Gobron, S., Chiba, N.: Simulation of peeling using 3D-surface cellular automata. In: Proceedings of the 9th Pacific Conference on Computer Graphics and Applications, pp. 338–347 (2001)
9. Iizuka, S., Endo, Y., Kanamori, Y., Mitani, J.: Single image weathering via exemplar propagation. Computer Graphics Forum **35**(2), 501–509 (2016)
10. Ishitobi, A., Nakayama, M., Fujishiro, I.: Visual simulation of weathering coated metallic objects. The Visual Computer **36**(10), 2383–2393 (2020)
11. Jain, N., Kalra, P., Kumar, S.: Simulation and rendering of pitting corrosion. In: Proceedings of the 2014 Indian Conference on Computer Vision Graphics and Image Processing, pp. 38:1–38:8 (2014)
12. Jeong, S., Kim, T.H., Kim, C.H.: Shrinkage, wrinkling and ablation of burning cloth and paper. The Visual Computer **27**(6), 417–427 (2011)
13. Jeong, S., Park, S.H., Kim, C.H.: Simulation of morphology changes in drying leaves. Computer Graphics Forum **32**(1), 204–215 (2013)
14. Kelager, M., Niebe, S., Erleben, K.: A triangle bending constraint model for position-based dynamics. In: Proceedings of the 7th Workshop in Virtual Reality Interactions and Physical Simulation, pp. 31–37 (2010)
15. Kimmel, B.W., Baranoski, G.V.G., Chen, T.F., Yim, D., Miranda, E.: Spectral appearance changes induced by light exposure. ACM Transactions on Graphics **32**(1), 10:1–10:13 (2013)
16. Kratt, J., Spicker, M., Guayaquil, A., Fiser, M., Pirk, S., Deussen, O., Hart, J.C., Benes, B.: Woodification: User-controlled cambial growth modeling. Computer Graphics Forum **34**(2), 361–372 (2015)
17. Merillou, S., Dischler, J.M., Ghazanfarpour, D.: Corrosion: Simulating and rendering. In: Proceedings of the Graphics Interface 2001 Conference, pp. 167–174 (2001)
18. Mérillou, S., Ghazanfarpour, D.: A survey of aging and weathering phenomena in computer graphics. Computers & Graphics **32**(2), 159–174 (2008)
19. Müller, M., Heidelberger, B., Hennix, M., Ratcliff, J.: Position based dynamics. Journal of Visual Communication and Image Representation **18**(2), 109–118 (2007)
20. O'Brien, J.F., Hodgins, J.K.: Graphical modeling and animation of brittle fracture. In: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, pp. 137–146 (1999)
21. Muñoz Pandiella, I., Bosch, C., Mérillou, N., Patow, G., Mérillou, S., Pueyo, X.: Urban weathering: Interactive rendering of polluted cities. IEEE Transactions on Visualization and Computer Graphics **24**(12), 3239–3252 (2018)
22. Paquette, E., Poulin, P., Drettakis, G.: The simulation of paint cracking and peeling. In: Proceedings of the Graphics Interface 2002 Conference, pp. 59–68 (2002)
23. Perlin, K.: Improving noise. ACM Transactions on Graphics **21**(3), 681–682 (2002)
24. Silling, S.: Reformulation of elasticity theory for discontinuities and long-range forces. Journal of the Mechanics and Physics of Solids **48**(1), 175–209 (2000)
25. Stomakhin, A., Schroeder, C., Chai, L., Teran, J., Selle, A.: A material point method for snow simulation. ACM Transactions on Graphics **32**(4), 102:1–102:10 (2013)
26. Stoney, G.G.: The tension of metallic films deposited by electrolysis **82**(553), 172–175 (1909)
27. Tanabe, R., Moriya, T., Morimoto, Y., Takahashi, T.: A generation method of rust aging texture considering rust spreading. In: Proceedings of International Workshop on Advanced Image Technology, pp. 399:1–399:4 (2015)
28. Turner, M.J., Clough, R.W., Martin, H.C., Topp, L.J.: Stiffness and deflection analysis of complex structures. Journal of the Aeronautical Sciences **23**(9), 805–823 (1956)
29. Wang, J., Tong, X., Lin, S., Pan, M., Wang, C., Bao, H., Guo, B., Shum, H.Y.: Appearance manifolds for modeling time-variant appearance of materials. ACM Transactions of Graphics **25**(3), 754–761 (2006)
30. Xue, S., Dorsey, J., Rushmeier, H.: Stone weathering in a photograph. In: Proceedings of the 22nd Eurographics Conference on Rendering, pp. 1189–1196 (2011)
31. Xue, S., Wang, J., Tong, X., Dai, Q., Guo, B.: Image-based material weathering. Computer Graphics Forum **27**, 617–626 (2008)

Appendix

We apply the triangle bending constraint in [14] to describe an angle constraint as a combination of simple length constraints to achieve high-quality simulations with fast convergence. As shown in Fig. A1(a), a triangle bending constraint controls the angle formed by two connected edges by setting a length constraint on the distance between the center of the triangle, which has the two edges as sides, and the junction point of the edges.

Figure A1(b) shows a tetrahedron bending constraint, which controls the angle formed by two adjacent faces. A tetrahedron bending constraint sets a length constraint on the distance between the center of a tetrahedron, which has the two faces as surfaces, and the edge shared by the faces. Notice we assume that each polygon has a shape similar to an equilateral triangle and the distance is equal to the distance between the center and midpoint of the edge. If the ends of the edges are $\boldsymbol{p}_1, \boldsymbol{p}_2$ and the other vertices are $\boldsymbol{p}_3, \boldsymbol{p}_4$, the middle point of the edge $\boldsymbol{v}$ and the center of the tetrahedron $\boldsymbol{c}$ are given as:

$$\boldsymbol{v} = \frac{1}{2}(\boldsymbol{p}_1 + \boldsymbol{p}_2), \quad \boldsymbol{c} = \frac{1}{4}(\boldsymbol{p}_1 + \boldsymbol{p}_2 + \boldsymbol{p}_3 + \boldsymbol{p}_4).$$

Therefore, if the target value of the distance between $\boldsymbol{v}$ and $\boldsymbol{c}$ is d_0, the tetrahedron bending constraint $C_{\text{tetrahedron}}$ is described as:

$$\begin{aligned} C_{\text{tetrahedron}}(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \boldsymbol{p}_4) &= \|\boldsymbol{v} - \boldsymbol{c}\| - d_0 \\ &= \frac{1}{4}\|\boldsymbol{p}_1 + \boldsymbol{p}_2 - \boldsymbol{p}_3 - \boldsymbol{p}_4\| - d_0. \end{aligned}$$

However, similar to the angle constraints described in Section 4.3, tetrahedron bending constraints must also be divided to distinguish between mountain and valley folds (Fig. A2). Given $l^1 = \|\boldsymbol{v} - \boldsymbol{p}_3\|, l^2 = \|\boldsymbol{v} - \boldsymbol{p}_4\|$, the target value of the angle ϕ_0 $(0 < \phi_0 < 2\pi)$, and the positions on the neutral axis $\boldsymbol{v}^1 = \boldsymbol{v} + l^1\boldsymbol{n}_{\text{neutral}}, \boldsymbol{v}^2 = \boldsymbol{v} + l^2\boldsymbol{n}_{\text{neutral}}$, two tetrahedron bending constraints $C^1_{\text{tetrahedron}}$ and $C^2_{\text{tetrahedron}}$ are defined as:

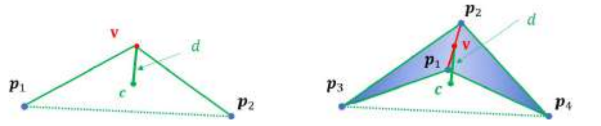
$$C^1_{\text{tetrahedron}}(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \boldsymbol{v}_1) = \frac{1}{4}\|\boldsymbol{p}_1 + \boldsymbol{p}_2 - \boldsymbol{p}_3 - \boldsymbol{v}^1\| - d^1,$$

$$C^2_{\text{tetrahedron}}(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_4, \boldsymbol{v}_2) = \frac{1}{4}\|\boldsymbol{p}_1 + \boldsymbol{p}_2 - \boldsymbol{p}_4 - \boldsymbol{v}^2\| - d^2,$$

where the constraint offsets d^1 and d^2 are given as:

$$d^1 = \frac{l^1}{4}\sqrt{2\left(1 + \cos\frac{\phi_0}{2}\right)}, \quad d^2 = \frac{l^2}{4}\sqrt{2\left(1 + \cos\frac{\phi_0}{2}\right)}.$$

Because $\boldsymbol{v}_1, \boldsymbol{v}_2$ are virtual points, their inverse masses are set to 0. Let $\Delta\boldsymbol{p}^1_1, \Delta\boldsymbol{p}^1_2$, and $\Delta\boldsymbol{p}^1_3$ be the corrections for $\boldsymbol{p}_1, \boldsymbol{p}_2$, and $\boldsymbol{p}_3$ according to $C^1_{\text{tetrahedron}}$, and let $\Delta\boldsymbol{p}^2_1, \Delta\boldsymbol{p}^2_2$, and $\Delta\boldsymbol{p}^2_4$ be the corrections for $\boldsymbol{p}_1, \boldsymbol{p}_2$, and $\boldsymbol{p}_4$ according to $C^2_{\text{tetrahedron}}$, respectively. If the inverse mass of $\boldsymbol{p}_i$ is denoted as w_i and the sums of the inverse masses $W = w_1 + w_2 + w_3 + w_4$, $W^1 = W - w_4$, and $W^2 = W - w_3$ are defined, then Eq. 3 yields corrections for the positions, given as:



(a) Triangle bending constraint [14]

(b) Tetrahedron bending constraint

Fig. A1: Comparison of two bending constraints. While the target of a triangle bending constraint (a) is the angle formed by the edges, the target of a tetrahedron bending constraint (b) is the angle formed by the faces

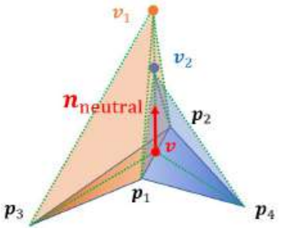


Fig. A2: Division of a tetrahedron bending constraint

$$\Delta\boldsymbol{p}^1_1 = -\frac{2u^1w_1}{W^1}(\boldsymbol{v} - \boldsymbol{v}^1), \quad \Delta\boldsymbol{p}^2_1 = -\frac{2u^2w_1}{W^2}(\boldsymbol{v} - \boldsymbol{v}^2),$$

$$\Delta\boldsymbol{p}^1_2 = -\frac{2u^1w_2}{W^1}(\boldsymbol{v} - \boldsymbol{v}^1), \quad \Delta\boldsymbol{p}^2_2 = -\frac{2u^2w_2}{W^2}(\boldsymbol{v} - \boldsymbol{v}^2),$$

$$\Delta\boldsymbol{p}^1_3 = +\frac{2u^1w_3}{W^1}(\boldsymbol{v} - \boldsymbol{v}^1), \quad \Delta\boldsymbol{p}^2_4 = +\frac{2u^2w_4}{W^2}(\boldsymbol{v} - \boldsymbol{v}^2),$$

where strains u^1 and u^2 are given as:

$$u^1 = 1 - \frac{d^1}{\|\boldsymbol{v} - \boldsymbol{v}^1\|}, \quad u^2 = 1 - \frac{d^2}{\|\boldsymbol{v} - \boldsymbol{v}^2\|}.$$

Considering that the ratio of corrections is equal to the ratio of the inverse mass if the constraint is not divided, we represent the corrections $\Delta\boldsymbol{p}_1, \Delta\boldsymbol{p}_2, \Delta\boldsymbol{p}_3$, and $\Delta\boldsymbol{p}_4$ as follows:

$$\Delta\boldsymbol{p}_1 = (\Delta\boldsymbol{p}^1_1 + \Delta\boldsymbol{p}^2_1) \times \frac{W^1W^2}{W(W^1 + W^2)},$$

$$\Delta\boldsymbol{p}_2 = (\Delta\boldsymbol{p}^1_2 + \Delta\boldsymbol{p}^2_2) \times \frac{W^1W^2}{W(W^1 + W^2)},$$

$$\Delta\boldsymbol{p}_3 = \Delta\boldsymbol{p}^1_3 \times \frac{W^1}{W},$$

$$\Delta\boldsymbol{p}_4 = \Delta\boldsymbol{p}^2_4 \times \frac{W^2}{W}.$$